

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of the claims:

1. (currently amended) A data service system, comprising:

a plurality of web servers, each servicing any request received by the data service system;

a plurality of application servers, each processing any request directed from any one of the web servers; and

a session state information managing system separate from the application and web servers and called by each of the application servers to allow different application servers to process requests belonging to a single session without requiring the requests to carry entire session state information, wherein the session state information managing system stores and distributes session state information to the application servers without requiring the application and web servers to store any session state information to process requests, and wherein the session state information managing system (1) maps the session state information to a sequence of bytes using a marshaling algorithm, (2) maps the sequence of bytes to a digest using a hash algorithm, and (3) maps the digest to a character string using an encoding algorithm to form a state reference.

2. (currently amended) The data service system of claim 1, wherein the session state information managing system further comprises

a session state information manager called by any one of the application servers when that application server processes a request of a session to (1) provide the session state information of the request to the application server and (2) generate the~~a~~ state reference for a new session state information for that request after the application server has processed the request and generated the new session state information for that request;

a store that stores all session state information received by the session state information manager.

3. (previously presented) The data service system of claim 2, wherein the state reference generated is unique to the corresponding session state information and is a fixed length character string.

4. (currently amended) The data service system of claim 1, wherein the sequence of bytes is mapped to the digest using a cryptographic hash algorithm 3, wherein the session state information manager generates the state reference by

~~mapping the session state information to a sequence of bytes using a marshaling algorithm;~~

~~mapping the sequence of bytes to a digest using a cryptographic hash algorithm;~~

~~mapping the digest to the character string using an encoding algorithm, thus forming the state reference.~~

5. (currently amended) The data service system of claim 1-4, wherein the marshaling algorithm further comprises a property that can un-marshall the sequence of bytes into a replica of the session state information without any loss of data.

6. (currently amended) The data service system of claim 1-4, wherein the digest mapped by the ~~cryptographic~~ hash algorithm is a fixed length sequence of bytes that is unique to the sequence of bytes.

7. (currently amended) The data service system of claim 1-4, wherein the encoding algorithm maps each byte in the digest to a hexadecimal representation, and creates a resulting character string.

8. (currently amended) A session state information managing system in a data service system having a plurality of duplicate application servers, each for processing requests, comprising:

a session state information manager called by any one of the application servers when that application server processes a request of a session to (1) provide session state information of the request to the application server and (2) generate a state reference for a

new session state information for that request after the application server has processed the request and generated the new session state information for that request;

a store that stores all session state information received by the session state information manager so the plurality of application servers are not required to store any session state information to process requests, wherein the session state information manager generates the state reference by:

mapping the session state information to a sequence of bytes using a marshaling algorithm;

mapping the sequence of bytes to a digest using a hash algorithm;

mapping the digest to the character string using an encoding algorithm, thus forming the state reference.

9. (previously presented) The session state information managing system of claim 8, wherein the state reference generated is unique to the corresponding session state information and is a fixed length character string.

10. (currently amended) The session state information managing system of claim 8, wherein the sequence of bytes is mapped to the digest using a cryptographic hash algorithm ~~9, wherein the session state information manager generates the state reference by~~

~~mapping the session state information to a sequence of bytes using a marshaling algorithm;~~

~~mapping the sequence of bytes to a digest using a cryptographic hash algorithm;~~

~~mapping the digest to the character string using an encoding algorithm, thus forming the state reference.~~

11. (currently amended) The session state information managing system of claim 8-10, wherein the marshaling algorithm further comprises a property that can un-marshall the sequence of bytes into a replica of the session state information without any loss of data.

12. (currently amended) The session state information managing system of claim 8-10,

wherein the digest mapped by the ~~cryptographic~~ hash algorithm is a fixed length sequence of bytes that is unique to the sequence of bytes.

13. (currently amended) The session state information managing system of claim ~~8-10~~, wherein the encoding algorithm maps each byte in the digest to a hexadecimal representation, and creates a resulting character string.

14. (currently amended) A method of allowing different application servers in a data service system to process requests belonging to a single session, comprising

(A) determining if a request starts a session;

if the request initiates the session, then

(B1) processing the request in one of the application servers and sending session state information of the request to a session state information manager separate from the application servers so the one of the application servers is not required to store the session state information for processing the request;

(B2) receiving a state reference unique to the session state information from the session state information manager and attaching the state reference to the response to the request;

if the request does not start the session, then

(C1) retrieving the session state information associated with the request from the session state information manager using a state reference contained in the request;

wherein the session state information manager generates the state reference by:
mapping the session state information to a sequence of bytes using a marshaling
algorithm;

mapping the sequence of bytes to a digest using a hash algorithm;

mapping the digest to the character string using an encoding algorithm, thus
forming the state reference.

~~(C2) processing the request with the retrieved session state information in one of the application servers and generating a new session state information;~~

~~(C3) sending the new session state information of the request to the session state information manager to receive a new state reference unique to the new session state information and attaching the state reference to the response to the request.~~

15. (original) The method of claim 14, wherein the step (C1) further comprises the steps of

verifying the session state information retrieved;

if the session state information is not verifiable, then performing the step (B1);

if the session state information is verified, then performing the step (C2).

16. (currently amended) The method of claim 14, wherein if the request does not start the session, then further:

(C2) processing the request with the retrieved session state information in one of the application servers and generating a new session state information;

(C3) sending the new session state information of the request to the session state information manager to receive a new state reference unique to the new session state information and attaching the state reference to the response to the request.
~~the session state information manager generates the state reference by~~

~~mapping the session state information to a sequence of bytes using a marshaling algorithm;~~

~~mapping the sequence of bytes to a digest using a cryptographic hash algorithm;~~

~~mapping the digest to the character string using an encoding algorithm, thus forming the state reference.~~